

(BinaryTrees) Reemplaça 0s per suma per sobre a profunditat parell X93188_ca

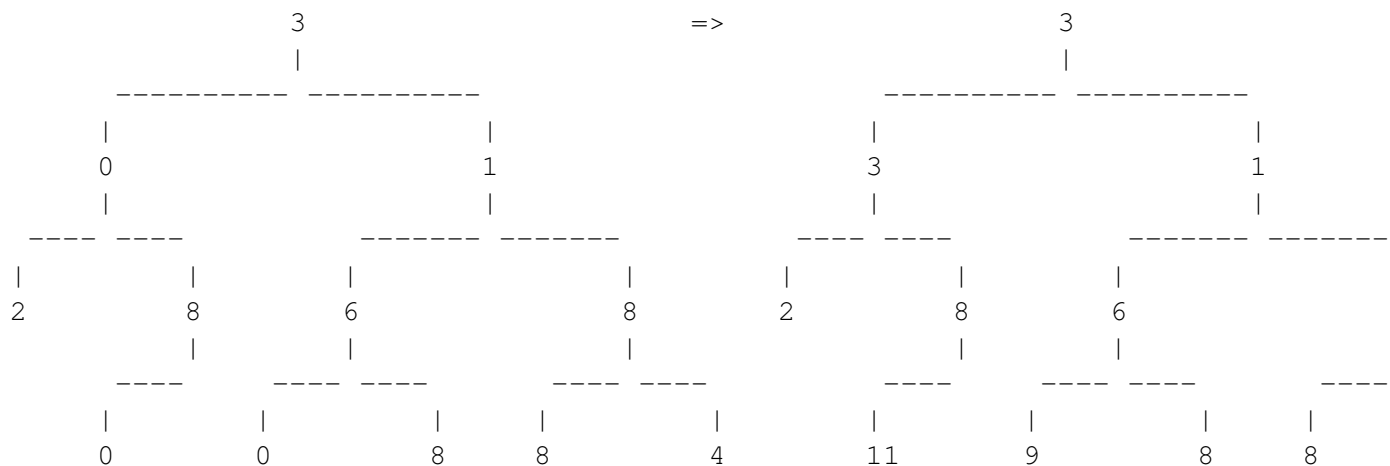
Implementeu una funció **RECURSIVA** que, donat un arbre binari de naturals, retorna un nou arbre que és idèntic a l'inicial, excepte que cada 0 s'ha reemplaçat per la suma dels elements a profunditat parell que apareixen per sobre d'aquest 0 (en l'arbre original), és a dir, les posicions a profunditat parell que són antecessores d'aquest 0.

Sobreentenem que l'arrel de l'arbre està a profunditat 0, els nodes directes des de l'arrel són a profunditat 1, els nodes a distància dos de l'arrel són a profunditat 2, i així successivament. Aquesta és la capçelera:

```
// Pre:  Sigui T el valor inicial de l'arbre t que es rep com a paràmetre.
//       Els valors guardats a T son majors o iguals a 0.
// Post: Sigui T' l'arbre retornat. T i T' tenen exactament la mateixa estructura
//       A més a més, per a cada posició p de T', si T té un valor x diferent d
//       llavors T' també té x a posició p.
//       En canvi, si T té valor 0 a posició p, llavors el valor de T' a posici
//       la suma de tots els valors de T a profunditat parell per sobre de p.
BinaryTree<int> replace0sWithAboveSumDepthEven(BinaryTree<int> t);
```

Aquí tenim un exemple de comportament de la funció:

```
replace0sWithAboveSumDepthEven(3(0(2,8(0,)),1(6(0,8),8(8,4)))) = 3(3(2,8(11,)),
```



Fixeu-vos que l'enunciat d'aquest exercici ja ofereix uns fitxers que haureu d'utilitzar per a compilar: Makefile, program.cpp, BinaryTree.hpp, replace0sWithAboveSumDepthEven.hpp. Només cal que creeu replace0sWithAboveSumDepthEven.cpp, posant-hi els includes que calguin i implementant la funció replace0sWithAboveSumDepthEven. I quan pugueu la vostra solució al jutge, només cal que pugueu un tar construït així:

```
tar cf solution.tar replace0sWithAboveSumDepthEven.cpp
```

Entrada

La primera línia de l'entrada descriu el format en el que es descriuen els arbres, o bé IN-LINEFORMAT o bé VISUALFORMAT. Després venen un nombre arbitrari de casos. Cada cas consisteix en una descripció d'un arbre un arbre binari de naturals. Fixeu-vos en que el programa que us oferim ja s'encarrega de llegir aquestes entrades. Només cal que imple-menteu la funció abans esmentada.

Sortida

Per a cada cas, cal escriure l'arbre binari resultant de cridar a la funció abans esmentada amb l'arbre d'entrada. Fixeu-vos en que el programa que us oferim ja s'encarrega d'escriure aquesta sortida. Només cal que implementeu la funció abans esmentada.

Exemple d'entrada 1

INLINEFORMAT

```

1(2(6(,(,)),8(0(,(9(2,4))),8(1,0(,(0))))))
2(6(7(3(,(5),1(,(3))),),8)
0(9(,(0(2,)),3(2,3(,(0))))
2(0,1(,(4(4,7))))
3(0(1(5(,(6),5),0(6(,(5),9(5,0))))),0)
9(4,0(,(0(,(0))))
6(4(3(7(,(2),3(,(4))),0),5(4(,(6(2,)),7(0
0(7(7(8,8),0(1(3))),)
0(0(,(0(,(3))),4(6,3)))
0(,(5(,(8(7(,(0),0(0,0))))))
5(8(3,),(9(,(0(2,9))))
9(6(5(5,),(3(,(0))),0)
0(0(9,4(0(3))),7)
1(3(,(0(1,))),)
6(1(9(0(8,8),),4(0,8(,(6))))),8(4,3))
0(0(4,9),7(3(6,7),0))
3(0(2,8(0,)),1(6(0,8),8(8,4)))
1(4(1,3),2(8,6(2,7(3,))))
9(2,9(8(1(5,),(1),2))
9(0,5(3(0,8),5)))

```

Exemple de sortida 1

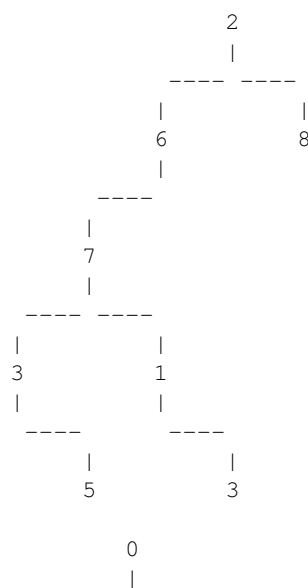
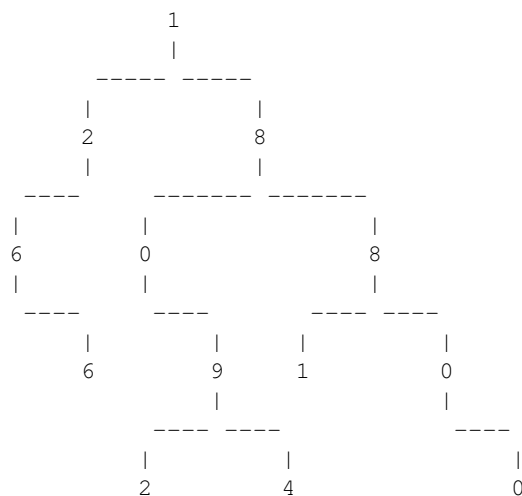
```

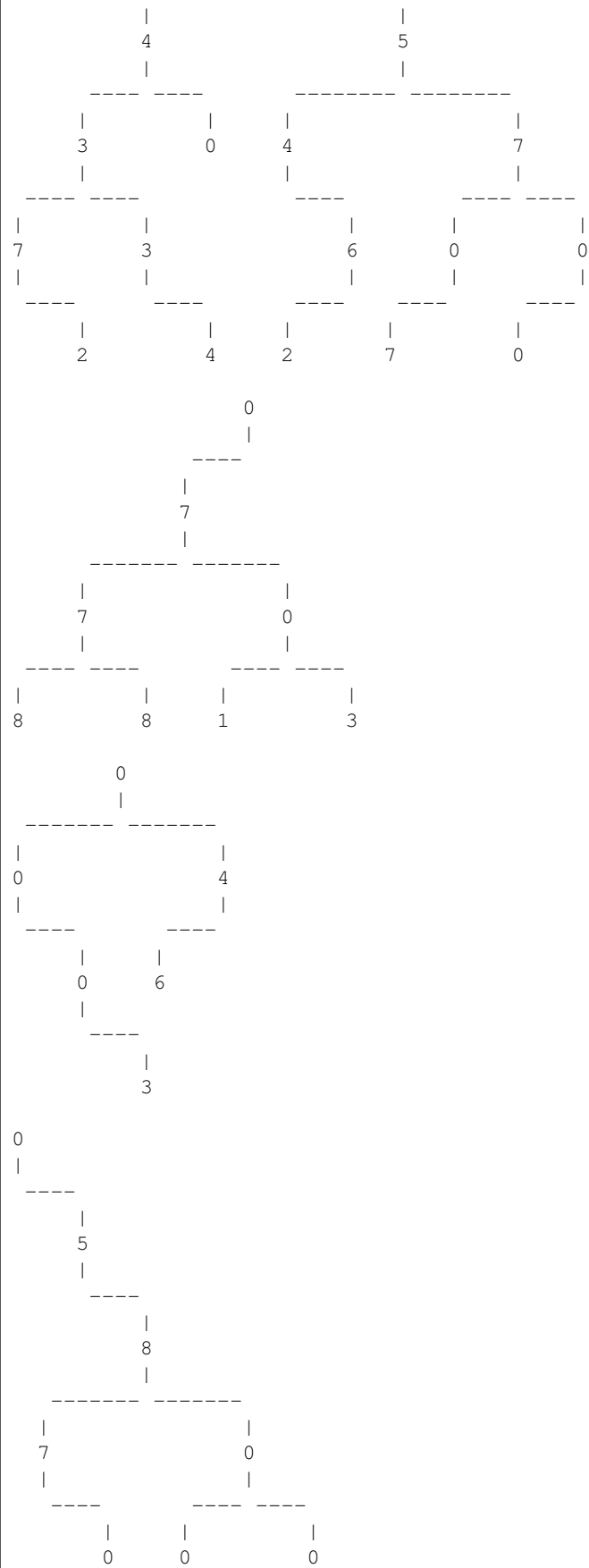
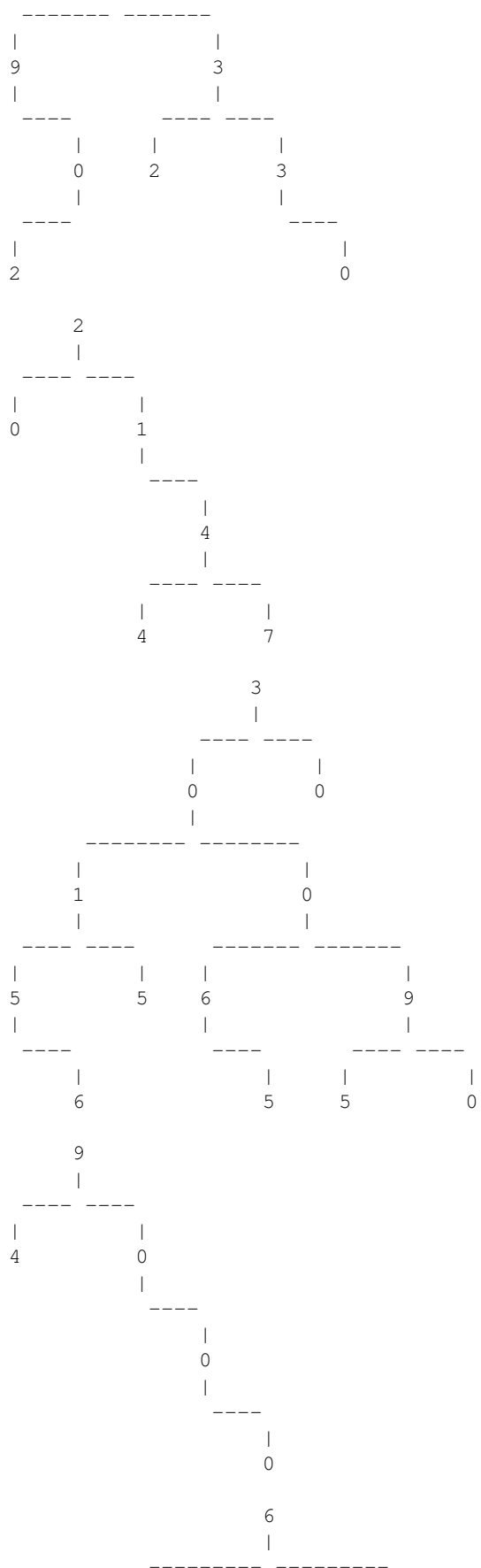
1(2(6(,6),),8(1(,9(2,4)),8(1,9(,9))))
2(6(7(3(,5),1(,3)),),8)
0(9(,0(2,)),3(2,3(,3)))
2(2,1(,4(4,7)))
3(3(1(5(,6),5),3(6(,5),9(5,3))),3)
9(4,9(,9(,9)))
6(4(3(7(,2),3(,4)),6),5(4(,6(2,)),7(13(7,),13(13,))))
,0(7(7(8),8),0(1,3)),)
0(0(,0(,3)),4(6,))
0(,5(,8(7(,8),8(8,8))))
5(8(3,),9(,5(2,9)))
9(6(5(5,),3(,12)),9)
0(0(9,4(4,3)),7)
1(3(,1(1,)),)
6(1(9(15(8,8),),4(10,8(,6))),8(4,3))
0(0(4,9),7(3(6,7),0))
3(3(2,8(11,)),1(6(9,8),8(8,4)))
1(4(1,3),2(8,6(2,7(3,))))
9(2,9(8(1(5,),1),2))
9(9,5(3(12,8),5))

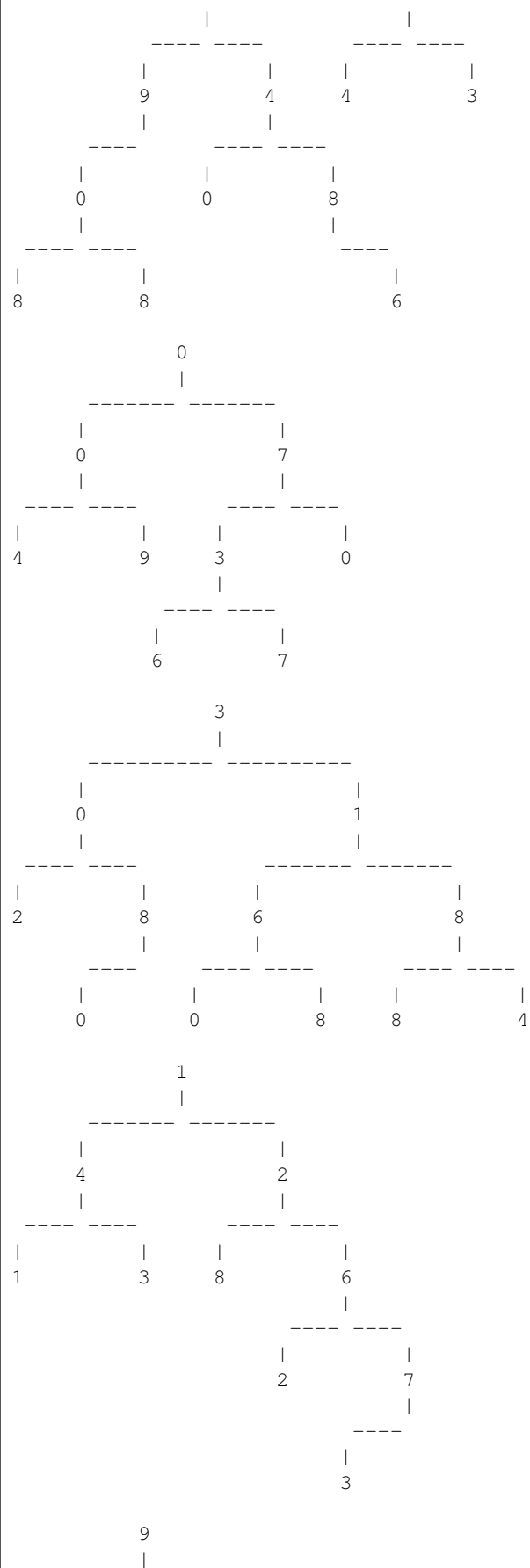
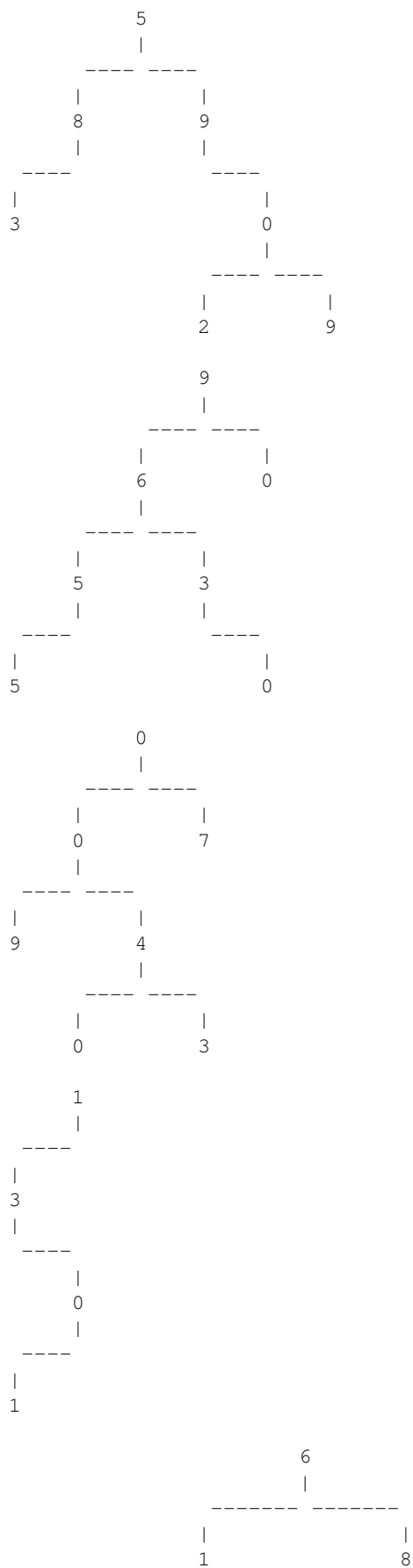
```

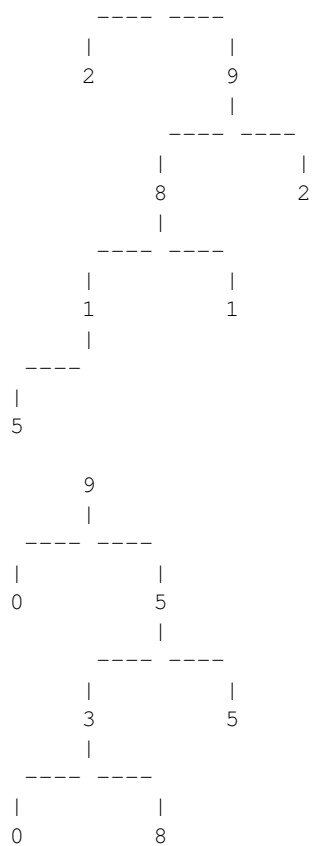
Exemple d'entrada 2

VISUALFORMAT

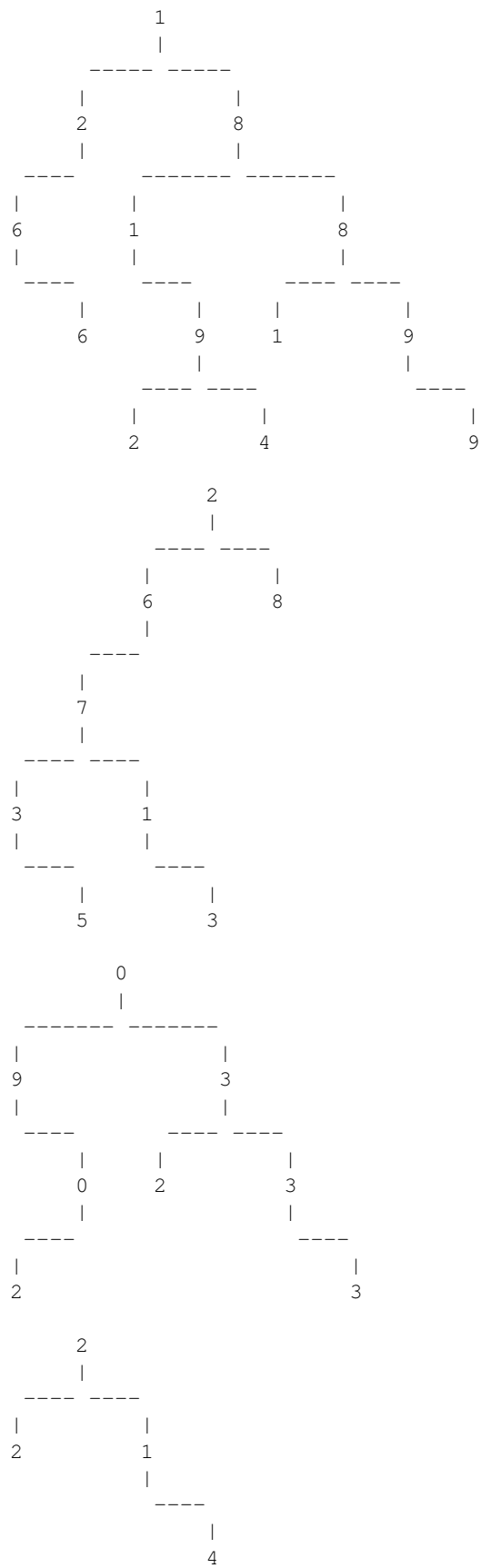


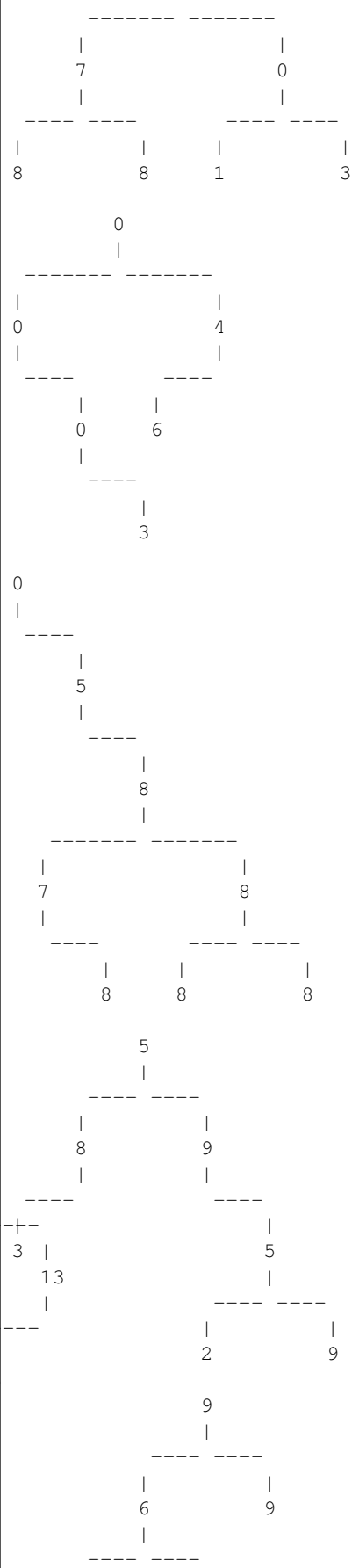
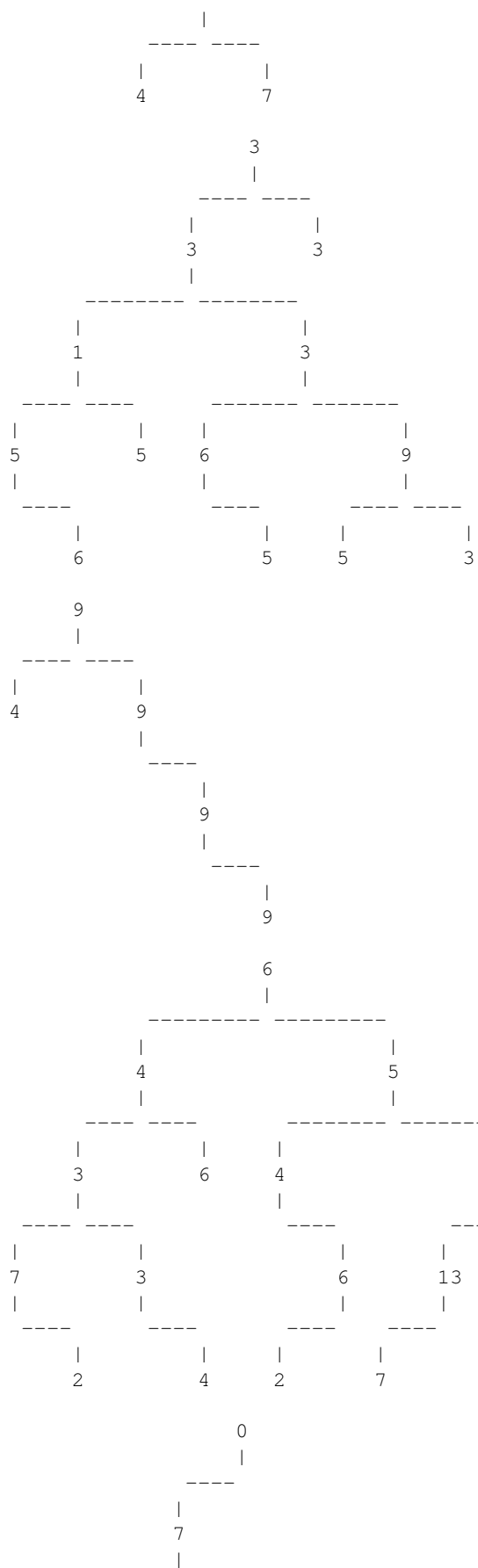


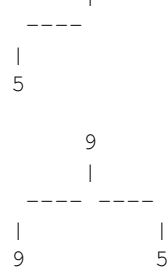
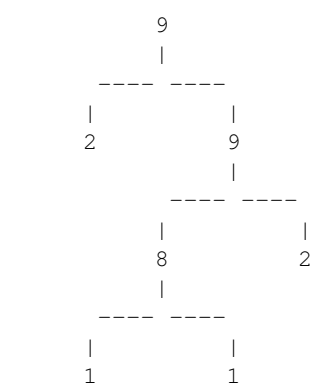
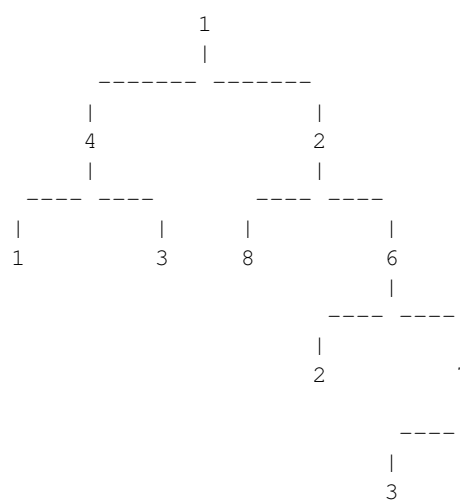
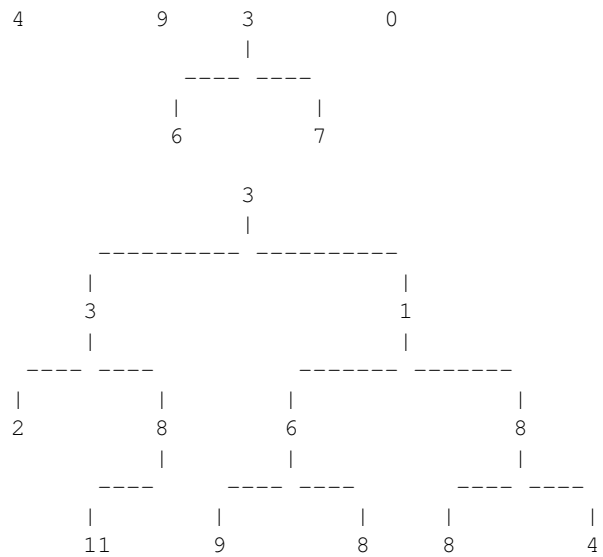
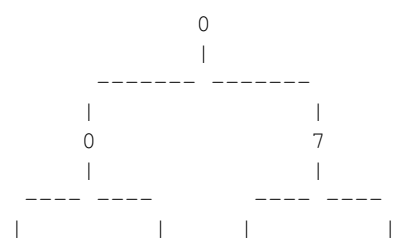
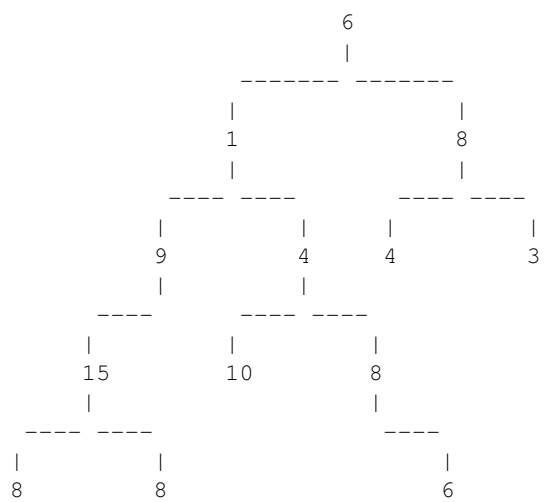
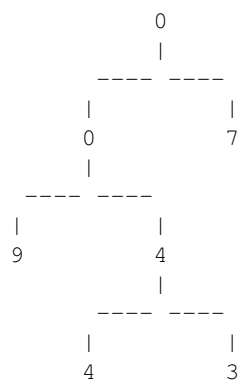
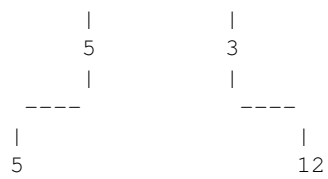


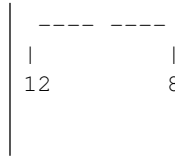
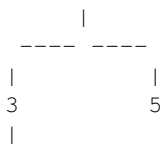


Exemple de sortida 2









Observació

La vostra funció i subfuncions que creeu han de treballar només amb arbres. Heu de trobar una solució **RECURSIVA** del problema. En les crides recursives, incloeu la hipòtesi d'inducció, és a dir una explicació del que es compleix després de la crida, i també la funció de fita/decreixement o una justificació de perquè la funció recursiva acaba.

Una solució directa superarà els jocs de proves públics i us permetrà obtenir una nota raonable. Però molt possiblement serà lenta, i necessitareu crear alguna funció recursiva auxiliar per a produir una solució més eficient capaç de superar tots els jocs de proves.

Avaluació sobre 10 punts:

- Solució lenta: 6 punts.
- Solució lenta + justificació: 8 punts.
- solució ràpida: 8 punts.
- solució ràpida + justificació: 10 punts.

Entenem com a solució lenta una que és correcta i capaç de superar els jocs de proves públics. Entenem com a solució ràpida una que és correcta i capaç de superar els jocs de proves públics i privats. La justificació val 2 punts i consisteix en definir correctament les PRE/POST de les funcions auxiliars que afegiu i en definir correctament les hipòtesis d'inducció i funcions de fita.

Informació del problema

Autoria: PRO1

Generació: 2026-01-25T21:33:47.856Z

© Jutge.org, 2006–2026.

<https://jutge.org>